

**ACTIVE DATA HIDING FOR  
SECURE ELECTRONIC MEDIA DISTRIBUTION**

**Background and Summary of the Invention**

The present invention relates generally to active data hiding, and more  
5 particularly, to method and system for robustly hiding active data into a host  
media data stream with errorless extractability.

Electronic media distribution imposes high demands on content  
protection mechanisms for secure distribution of media. Average users are  
starting to access and will soon be looking forward to purchasing multimedia  
10 content through the Internet. This urges the development of secure content  
distribution technologies with which content owners will agree to electronic  
distribution of digital media such as video and audio. The problem is  
amplified by the fact that the digital copy technology such as DVD-R, DVD-  
RW, CD-R, and CD-RW are widely available. Accordingly, imperceptible data  
15 hiding is becoming an attractive research area.

Previous research in the area of data hiding has been concentrated on  
passive data hiding, such as digital watermarking, for copyright protection or  
copy control. Passive data, as its name implies, can only be acted upon. In  
other words, passive data cannot actively perform a task. Key renewal or  
20 surveillance are two exemplary techniques for providing secure content  
distribution. In the case of passive data hiding, this type of functionality can  
only be achieved through additional functions built into the players. This  
greatly limits the application domain and the renewability of the system when  
additional functions are not available to the multimedia player devices.

Therefore, it is desirable to provide a method and system that can robustly hide active data into host media data stream with errorless extractability. Compared to conventional passive data hiding, active data hiding can improve renewability, controllability, and interoperability, provide additional application values and a higher level of security to electronic distribution of multimedia content.

For a more complete understanding of the invention, its objects and advantages refer to the following specification and to the accompanying drawings.

#### **Brief Description of the Drawings**

Figure 1 is a block diagram depicting an electronic media distribution system in accordance with the present invention;

Figure 2 is a flow diagram illustrating a method for hiding active data in accordance with the present invention;

Figure 3 is a flow diagram illustrating a method of decoding a host data signal embedded with an active data stream in accordance with the present invention; and

Figure 4 is a diagram depicting a perceptual mask in accordance with the present invention.

## **Detailed Description of the Preferred Embodiments**

Data hiding is generally defined as imposing a meaningful, imperceptible and extractable data stream onto a host signal. Imperceptibility and extractability are two technical criteria for conventional data hiding.

5 Imperceptibility means that the embedded data needs to be hidden into the host data signal such that it will not interfere with the quality (e.g., visibility or audibility) of the host signal. In addition, the embedded data needs to be extractable from the host signal on a player device. The extracted hidden data can then be used for copy control, copyright protection and other  
10 purposes.

In accordance with the present invention, active data hiding is a technique for hiding an applet or some other executable file into a host data signal. In addition to the imperceptibility and extractability requirements, active data hiding bears additional technical requirements. First, the size of  
15 the active hidden data is usually at least several hundred bytes. Instead of low bit rate embedding as in the case of conventional passive data hiding, active data hiding requires high bit rate embedding. However, for a fixed size host signal, it is more difficult to hide additional hidden data into the host signal, and thus it is more difficult to satisfy the imperceptibility requirement.

20 Second, active data hiding requires blind detection capability for electronic media distribution applications. Since only the protected medium, is available to the playing device, the extraction of any hidden data has to be performed without the original host medium. Third, due to the sensitivity to errors in an executable file, the extracted active hidden data has to be virtually  
25 errorless, i.e., the embedding has to be lossless.

An electronic media distribution system 10 is depicted in Figure 1. The media distribution system 10 includes a content provider device 12 that is connected via a distribution channel 14 to at least one player device 16. In operation, the original multimedia content is embedded with hidden data on the content provider device 12. The embedded media is then transmitted through the distribution channel 14 to the player device 16. At the player device 16, the embedded multimedia content may be played or used. In addition, the hidden data may be extracted from the embedded data signal.

In accordance with the present invention, a method for hiding active data in a host signal is shown in Figure 2. The host data signal is defined as original multimedia content, such as a digital video or audio signal. A preferred embodiment of the method uses a three-pass architecture to hide active data into a host data signal.

First, the host data signal is evaluated 22 to determine the media units of the host data. For a digital video signal, the media unit is one or more frames of video data. The host data signal may be further evaluated to determine the type of features associated with each media unit. For instance, a frame of video data includes features such as objects, texture regions and background. This information is subsequently used to determine how to embed the hidden data into the host signal.

The host data signal is then embedded with active hidden data, thereby forming an embedded data signal. Active hidden data is defined as a set of executable machine instructions, such as a JAVA applet or some other executable file or program. In order to embed the active data, the active data stream is mapped 24 into a sequence of binary data. Although in the case of

a JAVA applet the active data stream is mapped into a sequence of binary data, in some instances it may not be converted to binary data. The bit stream of binary data is then inserted imperceptibly into the host signal. It is also envisioned that the bit stream may be scrambled prior to insertion into the host signal. Thus, the embedded data signal designates a modified version of the host data signal that has additional meaningful data embedded into it. Although the invention is not limited to a particular embedding scheme, base domain embedding and spectrum domain embedding are two exemplary embedding schemes.

The host data signal may also be embedded with hidden control data. Hidden control data is used to govern the use of the active hidden data. For example, hidden control data may include synchronization data, identification data, access control data, keys, management data, error correction data, authentication data or other types of control data. These various types of control data are useful in the proper extraction of the active data stream as well as to control proper usage of the active data stream and the host signal. As will be more fully explained below, hidden control data is particularly useful to ensure errorless extraction of the active hidden data from the embedded data signal.

An additional embedding step is needed for each type of hidden control data embedded into the host signal. For illustration purposes, two types of control data are embedded into the host data signal in Figure 2: error correction data and authentication data. After generating the hidden control data in step 28, error correction data is first embedded into the embedded data signal. Subsequently, authentication data can be embedded into the

resulting data signal, thereby forming the embedded data signal that is to be transmitted to the player device.

Prior to being embedded into the host data signal, the active data stream may optionally be encrypted as shown at step 25. In this case, if the decryption key needs to be transmitted along with active data stream, the key may also be embedded in the control data.

Once the embedded data signal is received on the player device, a decoding process occurs as shown in Figure 3. As will be apparent to one skilled in the art, corresponding decoding techniques are performed to extract the embedded data signal received by the player device.

In this case, the authentication data is first extracted from the embedded data signal. An authentication check is performed to verify the reliability of the data signal. The active hidden data can be extracted from the embedded data signal.

Error correction data facilitates the extraction process of the active hidden data. Due to the additional control data hidden in the data signal, the detector/extractor on the player device can determine if there are any errors in the extracted active hidden data, and if so can further correct the errors such that the active hidden data is executable on the player device. The error correction process is shown at step 46. In this way, the present invention ensures errorless extractability of the hidden data.

At this point, the active hidden data can be executed on the player device. Again, the active data stream may optionally be decrypted prior to being executed on the player device.

In contrast to conventional passive data hiding, active hidden data introduces new functionality for ensuring secure electronic media distribution. For instance, an active data stream can be configured to permit feedback of information back to the content provider. In this case, when streaming or  
5 online preview is performed over the distribution channel (e.g., the Internet) to the player device, the information is transmitted back to the content provider or the content distributor.

In other instances, the active data stream may be configured to allow a play-once-preview, to enable renew keys or other management rules, or to  
10 scramble the host signal to prevent further unauthorized use of the content. These functions may be performed with the assistance of the hidden control data. For example, if an identification check or access control check fails, the host signal may be scrambled to prevent unauthorized use; otherwise the active data stream may perform other tasks while allowing authorized  
15 playback/usage of the host signal.

A methodology for hiding active data in an audio signal is presented to further illustrate the principles of the present invention. In this case, a three-pass, multi-layer approach is used to embed active hidden data, error correction data and authentication data into an audio signal.

A first pass embeds the active hidden data into the host data signal. Proper usage of the perceptual model ensures the imperceptibility of the embedded hidden data. The perceptual model takes advantage of human auditory system's inability to distinguish noise under conditions of auditory masking. That is, the presence of a strong audio signal makes a temporal or  
20 spectral neighborhood of weaker and imperceptible audio signals. Empirical  
25

data shows that the human ear cannot distinguish the differences when a minor change is made on a singular point or maskee point (under the condition it is still a maskee point before and after the modification), where a singular point, masker point and maskee point are defined as follows:

- 5       • a singular point  $I(j)$  is defined as iff  $\text{sign}(I(j)) = -\text{sign}(I(j-1)) \ \& \ \text{sign}(I(j)) = -\text{sign}(I(j+1))$ ;
- a masker point  $I(j)$  is defined as a point with an intensity value larger than a threshold  $\delta$ , i.e.,  $\text{amp}(I(j)) \geq \delta$ ;
- a maskee point  $I(j^k)$  is defined as a point that is under the mask of a  
10       masker point  $I(j)$ , i.e.,  $\text{amp}(I(j^k)) \leq \text{mask}(\text{amp}(I(j)))$

To illustrate the above-described principle, a perceptual mask is graphically depicted in Figure 4. In this figure, sample a is a masker point and samples b, c and d are maskee points. While the following description applies the perceptual model to an audio host signal, it is readily understood that the  
15       application of the perceptual model varies depending on the type of host data.

Furthermore, the application of the perceptual model also varies based on the particular embedding scheme being used to hide the active data. For instance, the masking ability of a given sample depends on its loudness in a base domain embedding scheme. In contrast, the masking ability of a given  
20       signal component depends on its frequency position and its loudness in the spectrum domain embedding scheme. Empirical results further show that the noise masking threshold at any given frequency is solely dependent on the signal energy within a limited bandwidth neighborhood of that frequency and at any given time is solely dependent on the signal energy within a limited  
25       temporal neighborhood. Accordingly, the base domain scheme has better



decoding performance in terms of speed than the spectrum domain scheme; whereas the spectrum domain scheme has higher survivability over compression than the base domain scheme.

As will be apparent to one skilled in the art, several techniques can be used to embed bits into the singular and maskee points of the host audio signal. For illustration purposes, a simple encoding technique is provided for embedding a sequence of bits  $Sb_1, Sb_2, \dots, Sb_M$  into the singular bits  $lsng_1, lsng_2 \dots lsng_M$ , of a host signal  $l_1, l_2, \dots l_n \dots l_N$ . The encoding technique is as follows:

- 10       •       If  $l(j)=0$ , set  $l(j)=l(j)+1$
- If the embedding bit  $Sb_m$  is 0 and the  $m$ th singular point is  $lsng_m$ , then set  $lsng_1$  to 0.
- If the embedding bit  $Sb_m$  is 1, then leave  $lsng_m$  unchanged or set  $\epsilon_1 \leq lsng_m \leq \epsilon_2$ , where  $\epsilon_1$  and  $\epsilon_2$  are lower and upper bound with  $\epsilon_2$  controlled by perceptual mark.
- 15

To ensure maximum detectability, error correction data and authentication data should be embedded into different data layers within the host data signal. The active hidden data layer and any subsequent control data layers are preferably orthogonal to each other. The orthogonality of the embedded layers avoids any interference between embedded bits, thereby ensuring extractability of each layer. For example, singular points and maskee points are two orthogonal features of the host data signal which may be used to hide different data layer. Accordingly, active data may be hidden in the singular points and the control data in the maskee points of the host signal. Alternatively, if the signal is partitioned into subsets or subspaces, then the features extracted from the different subsets or subspaces will be

orthogonal to each other. Thus, it is envisioned that other orthogonal aspects of the host data signal, such as other orthogonal features in the same domain (e.g., time, spectrum, etc.) or other features extracted from different orthogonal domains, may be used to embed the different layers. Although  
5 orthogonality is preferred, it should be noted that different data layers may also be non-orthogonal as far as the zero false rate is guaranteed for the extraction of the active data stream.

Next, error correction data is embedded into the host data signal. Again, the error correction data is hidden in a second orthogonal layer of the  
10 host signal. For illustration purposes, a 2D checksum error correction technique is being used to embed error correction data. Assume the error correction bit number is  $Q$  and the active data stream bit number is  $M$ . Thus, the error correction stream length (number of bits) satisfies  $M=(Q/2)^2$  for the 2D checksum technique. For example, an active data stream having a length  
15 of 4000 bits requires only  $64 \times 2 \approx 128$  error correction bits in the case of 2D checksum. An exemplary 2D checksum technique is provided as follows:

- Let  $Q = \text{ceiling}[2M^{1/2}]$ , i.e., let  $Q$  be the smallest integer which is no less than  $2M^{1/2}$ .
- Arrange  $S_b = S_{b_1}, S_{b_2}, \dots, S_{b_M}$  into  $Q/2$  chunks  
20  $SB(1) = SB(1)_1, SB(1)_2, \dots, SB(1)_{Q/2} = S_{b_1}, S_{b_2}, \dots, S_{b_{Q/2}},$   
 $SB(2) = SB(2)_1, SB(2)_2, \dots, SB(2)_{Q/2} = S_{b_{Q/2+1}}, \dots, S_{b_Q} \dots$  and  
 $SB(Q/2) = SB(Q/2)_1, SB(Q/2)_2, \dots, SB(Q/2)_{Q/2} = S_{b_{(Q-2Q)/4+1}}, \dots, S_{b_M}$
- Let  $E_q = \text{LSB}(SB(q)_1 + SB(q)_2 + \dots + SB(q)_{Q/2})$  for  $q \in (1, Q/2)$  and  
25  $E_q = \text{LSB}(SB(1)_q + SB(2)_q + \dots + SB(Q/2)_q)$  for  $q \in (Q/2, Q)$ ,  
where  $\text{LSB}(S)$  denotes the least significant bit of  $S$ .

While the above-described example employs a 2D checksum error correction technique, it is readily understood that other error correction techniques are within the scope of the present invention, including but not limited to Perfect

codes, Quasi-perfect code, Hamming code, Duel codes, Hadamard codes, Golay codes, Nordstrom-Robinson codes, BCH codes, Cyclic codes, MDS codes, Reed-Muller codes, Kerdock codes, Preparata codes, Quadratic-residue codes, Reed-Solomon codes, and Justesen codes.

5           Lastly, authentication data is embedded into the host data signal. Again, the authentication data is placed into a third orthogonal layer of the host signal. In this case, a preferred authentication scheme places the authentication value into the least significant bit of each sample of the host audio signal. To ensure orthogonality,  $\varepsilon_1$  shall be set to 2 or larger for both  
10 singular point and maskee point embedding of the authentication data. A overview of the authentication algorithm is as follows:

- 15           • Choose verification block size B and dependent block size D (for example, B=128 and D=512 bits). Assume the host signal is a 16bits audio, concatenating all the high bits (all the bits except the least significant bit) of the 512 samples yields a message Mb of 15x512=7680bits. By further concatenating a key of 512bits (or a key of shorter length which is padded to 512bits (or a key of shorter length which is padded to 512bits), a message MB of 8192bits is produced.
- 20           • Computer the one way hash with the MD5 algorithm,  $MB'=h=H(MB)$  to generate a 128 bit message MB'. (Append time or other secondary hidden data, such as the error correction bits, host signal length, and/or owner information, if B> 128 bits.)
- 25           • Use public key (or secret key, depends on different applications) cryptography method to sign MB' with secret key K creating  $MB''=Sgn(K, MB')$ .
- 30           • Insert the B bits message, MB'', into the least significant bit of each sample, from 1→0 if embedding 0 or 0→1 if embedding 1, into the verification block.

